Factors Influencing the Effort of EAI Projects – A Repertory Grid Investigation

Completed Research Paper

Holger Wagner
KMA - Umwelttechnik GmbH
Eduard-Rhein-Straße 2
53639 Königswinter, Germany
h.wagner@kma-filter.de

Oleg Pankratz
University of Cologne
Albertus-Magnus-Platz
50923 Cologne, Germany
pankratz@wiso.uni-koeln.de

Dirk Basten
University of Cologne
Albertus-Magnus-Platz
50923 Cologne, Germany
basten@wiso.uni-koeln.de

Werner Mellis
University of Cologne
Albertus-Magnus-Platz
50923 Cologne, Germany
mellis@wiso.uni-koeln.de

Abstract

Today's enterprises often face heterogeneous application landscapes. Many of those companies struggle with effective and efficient accomplishment of enterprise application integration (EAI), which results in significant time and budget overruns. As regards EAI project management, a major reason for failure is considered to be underestimation of effort. The underestimation has been found to be an aftermath of applying estimation methods that do not account for all relevant factors influencing EAI project effort. We therefore explore factors affecting the effort of such projects in this study. Applying Repertory Grid, we conduct 22 semi-structured expert interviews. 91 factors influencing the effort of EAI projects in nine categories emerge from these interviews. We provide an extensive overview of effort-influencing factors and their classification, which can be used as a checklist in EAI projects. Future research can additionally use our findings as basis for development of more accurate effort estimation models.

Keywords: Enterprise application integration, Effort estimation, Repertory grid, Project management

"The dynamism of the software field means that the software estimation discipline needs to be continually reinventing itself." (Boehm and Fairley 2000, p. 24)

Introduction

Commonly referred to as enterprise application integration (EAI), the integration of enterprises' applications involves the consolidation of logically related application systems (Themistocleous and Irani 2002). EAI aims at automating the interaction among them in order to enable the support of business processes across existing systems. The systems are usually not changed and continue to exist independently (Linthicum 2004). Typical examples of EAI include providing services for other systems and data exchange across systems.

Nowadays, large enterprises are often confronted with heterogeneous application landscapes (Ho and Lin 2004; Khoumbati and Themistocleous 2006), that is, they run and maintain hundreds or even thousands of applications in parallel (Lam 2005; Riempp and Gieffers-Ankel 2007). Accordingly, integrating applications for the purpose of consistent support of business processes is commonly in focus of today's enterprises and among the major challenges for Chief Information Officers (Thompson 2007). Extensive budget for the development and maintenance of interfaces (Crosman et al. 2007; Ruh et al. 2000) and the growing market for integration solutions emphasize the significance of EAI in modern enterprises (Gartner Inc. 2006, 2008). As Gericke et al. (2010, p. 678) outline, "[e]ffective and efficient integration of applications is, therefore, an important, relevant problem".

However, many enterprises struggle with the accomplishment of EAI (Tuft 2001), leading to significant time and budget overruns (Rosa et al. 2013; Tuft 2001). A major reason for EAI project failure is considered to be underestimation of the required effort (Lublinsky and Farrell 2002), which shifts the focus to project management. Underestimation of effort has been found to be an aftermath of applying estimation approaches which have been designed for development of new, individual systems and are thus unsuitable to estimate the effort required for integration of multiple, already existing systems (Rosa et al. 2013). In these approaches (e.g., Ganly, 2009; Seaver, 2006), existing descriptions of how to estimate EAI project effort leave many parameters up to managers to decide, who typically lack adequate data (Rosa et al. 2013). Moreover, limited understanding of the nature of ERP systems in contrast to traditional systems leads to "the use of [...] questionable estimation techniques" which focus on software issues while neglecting, for instance, organizational impacts (Rosa et al. 2013, p. 544). As a consequence of applying the existing estimation approaches, "the ability to obtain accurate cost estimates remains somewhat problematic" (Rosa et al 2013, p. 539). In this context, the underestimation of required effort indicates that not all factors influencing the effort are considered in the estimation process. It is therefore primarily essential to explore factors affecting the effort of such projects. In this study, we thus set out to answer the following research question:

What factors need to be considered in effort estimation of EAI projects?

We intend to answer this question by eliciting knowledge of EAI experts. We apply the Repertory Grid Technique (RepGrid) (Tan and Hunter 2002), which has been widely and successfully used in the information systems discipline (Napier et al. 2009; Siau et al. 2010; Tan and Gallupe 2006; Whyte and Bytheway 1996). As RepGrid enables the elicitation of explicit and tacit knowledge, it is suitable for investigating the knowledge of experts and provides insights of high scientific value (Stewart et al. 1981).

Our study contributes to research and practice as follows. While existing estimation approaches provide only limited accuracy, a comprehensive list of effort-influencing factors promises a more extensive and adequate consideration of the variety of effort drivers that need attention in the EAI context. Experts can apply our factors overview as a checklist; our results are thus of high relevance primarily for the practice. Since many factors are identified that are not considered in existing models, future research can additionally use our findings as basis for development of more accurate effort estimation models.

The paper proceeds as follows. The next section provides an overview of previous work on EAI and effort estimation of EAI projects. Subsequently, we describe and argue for the concrete design of our research method, followed by our results and discussion. Our article ends with a short conclusion.

Related Work

Enterprise Application Integration

Various denotations have been used to describe the integration of application systems in the information systems literature (Themistocleous and Irani 2002). Examples include application integration (Sprott 2000), systems integration (Hasselbring 2000), value chain integration (Yang and Papazoglou 2000), extended business integration (Markus 2000), and, most commonly, EAI (e.g., Lam 2005; Linthicum 2000; Zahavi 2000). EAI can be defined as consolidation of different application systems that are logically related components of enterprise information systems (Linthicum 2004; Themistocleous and Irani 2002). In contrast to new developments of application systems, EAI refers to subsequent linkage of already existing systems. The purposes of EAI are the automation of interactions between application systems of one or multiple enterprises and the support of business processes across systems. Examples of such interactions include access to functions of different systems and data interchange across application systems. The corresponding systems are preferably not changed and remain functioning independently.

EAI projects are considered resource-intensive and critical for enterprises' success (Gericke et al. 2010; Lam 2005; Rosa et al. 2013; Themistocleous and Irani 2002). Despite their importance, EAI projects often fail according to criteria of time and budget (Rosa et al. 2013; Tuft 2001). Striving to improve this situation, scholars have conducted numerous studies to identify success factors of EAI projects (e.g., Ho and Lin 2004; Lam 2005; Mendoza et al. 2006; Schwinn and Winter 2007; Themistocleous 2004; Zaitun 2001). In their review, Gericke et al. (2010) identify characteristics that the majority of these studies have in common, concluding that most studies focus on an intra-organizational rather than inter-organizational context and follow a project-oriented rather than process-oriented view. The latter finding indicates that EAI is typically conducted in projects rather than seen as a continuous process. Since a new EAI solution can additionally be seen as a *change* project, many success factors pertain to topics like technology adoption and overcoming resistance to change (Gericke et al. 2010).

However, a closer look at the research of EAI project success factors reveals an interesting finding. Despite EAI being a complex technological task (Mendoza et al. 2006), one recurrent theme pertains to EAI project management (Ho and Lin 2004; Lam 2005; Mendoza et al. 2006; Themistocleous 2004; Zaitun 2001). Themistocleous (2004), Zaitun (2001), and Mendoza et al. (2006) see project management in general as a success factor of EAI projects. Ho and Lin (2004) more precisely consider the expertise of the project manager to be crucial for success. Lam (2005) additionally finds support for several planning-related factors, such as required planning skills and expertise and appropriate budget calculation. Considering the high time and budget overruns in EAI projects (Rosa et al. 2013; Tuft 2001), the relevance of project management is plausible. As budgets are typically calculated based on effort estimates (Basten and Sunyaev 2014), a promising approach to avoid EAI project failure is to scrutinize effort estimation in such projects (Rosa et al. 2013).

Effort Estimation of EAI Projects

Effort estimation plays a crucial role in software projects in general (Basten and Sunyaev 2014) and EAI projects in particular (Rosa et al. 2013). Effort estimates are used for project planning and staffing, success assessment, progress monitoring, and evaluation of developers (Moløkken-Østvold et al. 2004). Although scholars in the software engineering and information systems discipline disagree on a definition of the term *effort estimate* and related processes (Grimstad et al. 2006), its general purpose can be considered an early and reliable prediction of the effort required to complete a project (Rosa et al. 2013). Despite extensive research dedicated to effort estimation, it is still unclear how to accurately estimate the effort, particularly for large, complex projects (Jørgensen 2014).

Rosa et al. (2013) deem effort estimation a major problem in the field of application integration due to the complexity of enterprises' application landscapes. The challenge of accounting for the complexity in effort estimates can be tied to the characteristics of enterprise systems. Although comprising multiple single applications, the entire system cannot be thought of as a bunch of individual applications in the context of effort estimation. More precisely, estimating the effort for integrating the entire system cannot be achieved by simply combining the effort estimated for integrating the individual applications. Besides software and hardware, overall systems comprise business processes and organizational structures, leading to an intertwined entity of well-acknowledged complexity (Stensrud and Myrtveit 1998). Despite recent attempts to reduce complexity in form of agile or tailorable systems, the prevailing failure rates of EAI projects indicate that the challenge of complexity has not yet been solved (Rosa et al. 2013). Rosa et al. (2013) also identify deficiencies in previous research on effort estimation in the EAI context. For instance, scholars either focus on single vendors only or do not provide suitable databases (e.g., Francalanci 2001; Stensrud 2001). Furthermore, studies can be found that concentrate on size and complexity of implementations (Daneva 2008, 2010), but still lack practical methods for estimating effort in a majority of implementations.

As regards effort estimation in general, Jørgensen et al. (2009) see a distinction between formal models and expert judgments. Estimates can be based either on one of these approaches or on a combination of the two. Both approaches attract a variety of studies to either develop and improve formal models (e.g., Boehm et al. 2000; Dejaeger et al. 2012; Heemstra and Kusters 1991; Kemerer 1987; Xia et al. 2008) or assess and advance effort estimation with expert judgments (e.g., Connolly and Dean 1997; Gray et al. 1999; Grimstad and Jørgensen 2007; Gruschke and Jørgensen 2008; Jørgensen 2004a), respectively. While formal models have a long research tradition beginning in the 1960s (Jørgensen et al. 2009), empirical findings show that such models are seldom applied (Moløkken-Østvold et al. 2004). One reason for practitioners' preference of expert estimation over formal models is that the former approach can be more easily applied while providing at least an equivalent level of accuracy (Basten and Sunyaev 2014). According to Jørgensen (2004b, p. 55), "there is no substantial evidence supporting the superiority of model estimates over expert estimates".

In conclusion, we believe that it is more promising for research to focus on expert judgments than formal models for two reasons. First, formal models are unlikely to account for peculiarities of most EAI projects and can thus be applied to specific projects only. Second, practitioners tend to rely on expert judgments as the predominant approach for effort estimation and thus depend on means to improve this course of action. While formal models follow a quantitative approach to master the challenge of effort estimation in EAI projects, we apply a qualitative approach to explore experts' knowledge on the factors that affect the EAI project effort. Thus, our study is in line with research emphasizing the criticality of expert knowledge for EAI projects (Ho and Lin 2004; Kugeler and Vieting 2003; Lam 2005; Themistocleous 2004; Themistocleous and Irani 2002). Aiming to capture a wealth of EAI expertise, we explore practitioners' experience by applying RepGrid, which is suitable to investigate both explicit and tacit knowledge and to provide insights of high scientific value (Stewart et al. 1981).

Research Design

Data Collection

RepGrid is a technique of conducting semi-structured interviews to explore people's personal construct systems. The clinical psychologist George Kelly argues that all humans observe their surroundings and develop personal construct systems, which they use to interpret events and make decisions (Kelly 1955). According to Kelly's (1955) personal construct theory (PCT), such a mental model of the world is composed of elements and constructs. While elements were people in the origins of PCT, researchers recognized that, depending on the context and aims of the investigation, elements can be any objects of people's thoughts like computers, organizations, or software projects (Smith 1986). Constructs represent elements' qualities and are used to distinguish between elements (Smith 1986). Such qualities may, for instance, be evaluative (useful, appropriate), physical (tall, beautiful), or character attributes (kind, guile).

An important characteristic of constructs is their bipolarity. As Fransella et al. (2004, p. 7) point out, people "never affirm anything without simultaneously denying something". Moreover, it is important not to equate constructs with their verbal labels – while constructs exist in people's minds, construct labels are means to describe and communicate constructs. This distinction is crucial since different people often put different labels on the same things and vice versa. According to Shaw and Gaines (1989), four possible semantic constellations exist: consensus (same terminology for same concepts), correspondence (different terminology for same concepts), conflict (same terminology for different concepts), and contrast (different terminology for different concepts). Therefore, a key factor for ensuring validity in qualitative research is being aware of potential semantic ambiguities and addressing them adequately (e.g., using Laddering as described below). An extensive discussion of PCT is given in Kelly (1955) or Fransella et al. (2004).

Application of RepGrid in qualitative investigations like ours involves comparisons of elements (in our study: EAI projects) to identify similarities/differences between them, thus eliciting the constructs (in our study: effort-affecting factors in these projects). In this manner, RepGrid has been widely and effectively applied in the information systems discipline. For example, Siau et al. (2010) recently used RepGrid to explore characteristics of software development team members; Napier et al. (2009) applied it to investigate skills of successful information technology project managers; and Tan and Gallupe (2006) took advantage of RepGrid to examine business and information technology thinking. A comprehensive overview and discussion of RepGrid design decisions can be found in Tan and Hunter (2002).

Two major challenges exist related to exploring individual knowledge. First, people are often unaware of possessing it. As Polanyi (1966, p. 4) points out, "We can know more than we can tell". This kind of knowledge is typically referred to as tacit knowledge (Nonaka 1994). Second and complicating the former, this tacit knowledge constitutes the larger part of personal knowledge – explicit knowledge is only the tip of the iceberg (Polanyi 1966). RepGrid overcomes these challenges since it allows exploring tacit knowledge by making it explicit (Jankowicz 2001; Stewart et al. 1981). RepGrid also brings further benefits. It is effective in identifying new constructs rather than confirming known ones, which serves our purpose to explore factors that affect EAI project effort. Furthermore, the prescribed way of interviewing experts reduces researcher bias as constructs emerge directly from the interviewees (Stewart et al. 1981).

Hinkle (1965) developed Laddering as an extension to RepGrid to account for the hierarchical relations between constructs. Laddering involves additional questions based on the identified construct, leading in the directions downwards, upwards, or sideways (Rugg et al. 2002). The primary focus of Downwards Laddering is clarification of construct meaning (Jankowicz 2004) by asking questions like "How could you tell that something was X?" or "Can you give me examples of X?". Upwards Laddering uncovers constructs on higher hierarchical levels by asking "Why do you prefer X?" and quickly leads to personal core beliefs of the respondent. Finally, Sideways Laddering helps to elicit further constructs on the same hierarchical level by asking "Can you think of more aspects like X?".

For our application of RepGrid, we chose a twofold approach to obtain interviewees for our expert group. First, we used a convenient sample by contacting members of the workgroup for service-oriented architectures of the German chapter of the Association for Computing Machinery. Application integration is one of the focuses of this workgroup. Second, we used the list for leading enterprises within the system integration and consulting domain in Germany (http://www.luenendonk.de) and contacted these enterprises via telephone to acquire further interviewees. Our final sample consisted of 22 EAI experts (cf. Table 1) from ten organizations (cf. *Appendix A* for their characteristics). Beside the respondents' current position, Table 1 provides their professional experience in three columns: job experience in years, number of projects (in brackets the number of projects on which they were involved in the planning phase), and industries in which the experts worked in their career.

Application of RepGrid requires multiple design decisions to be made, in compliance with study aims and context. Table 2 provides the numerous design alternatives of RepGrid along with our choices. The list of design alternatives is collected from multiple seminal works on designing RepGrid studies and Laddering (e.g., Fransella et al. 2004; Rugg et al. 2002; Stewart et al. 1981; Tan and Hunter 2002). In accordance with our RepGrid design choices, we interviewed the 22 EAI experts as follows (cf. also Figure 1).

At the beginning of each interview, we collected demographic information from our respondents and briefly explained PCT and RepGrid to make them familiar with the interview technique. Subsequently, interviewees identified elements by naming four EAI projects from their experience. We asked our respondents to select projects of which they had a good recollection. Following the purpose of our study to identify a broad range of constructs, we did not further restrict our respondents in selecting elements; thus, we are confident to have collected a representative sample of EAI projects and a broad set of effort-affecting factors in such projects. The project names were written on index cards, including ranks regarding the integration effort. At this point, audio-recording was started – we recorded and transcribed all interviews with permission from our respondents to capture all relevant information. Subsequently, two of the projects (dyads) were compared regarding factors that affected the effort of these projects. We used the effort ranks to compare projects with the most and the least effort first, which allowed us an 'easy' start for the first dyad. Respondents identified factors (raw constructs) by naming differences between the compared EAI projects. Once a construct was identified, Downwards Laddering was applied to deepen the understanding and counteract semantic ambiguities. The interviewer documented all constructs on additional paper cards during the interviews to provide a permanently visible representation of the respondents' construct systems. We continued eliciting constructs for the chosen dyad until the interviewees could not think of further constructs. Then, the interviewer chose another pair of projects for comparison. This in turn was repeated until all dyads were considered and the respondents indicated that the emerged construct systems properly represented their personal view on factors affecting the effort of EAI projects. In the closing phase, we asked the respondents to characterize the considered projects (cf. *Appendix B* for an overview).

Table 1. Interviewee Characteristics

No.	Current position	Experience		
		Job	Projects (planning)	Industries throughout the career
1	Head of consulting	15	10 (7)	Telecommunication
2	Manager	10	8 (4)	Logistics
3	Business developer manager	25	5 (5)	FS, S&T, logistics
4	Manager	7	4 (4)	FS
5	Solution manager EAI	12	5 (3)	FS, telecommunication
6	Project manager	7	3 (2)	FS
7	Solution architect	25	4 (3)	S&T, public sector, telecommunication
8	Practice area leader	10	25 (15)	FS, S&T, logistics, health
9	Solution architect	25	3 (0)	FS, media
10	Senior consultant, architect	13	23 (18)	FS, telecommunication, traffic, health
11	Senior system architect	8	4 (3)	FS, automotive
12	Chief system architect	25	10 (8)	FS, automotive, S&T, logistics, defense
13	Architect, project manager	10	6 (3)	FS, Public sector, media
14	Project lead	6	5 (4)	FS, automotive, pharmaceutics, logistics
15	Senior consultant, architect	7	10 (3)	FS, automotive, health, S&T
16	Architecture consultant	18	6 (6)	Pharmaceutics, telecommunication
17	Project manager	20	4 (4)	FS, telecommunication, logistics
18	Partner	10	5 (4)	FS
19	Consultant, architect	15	6 (4)	FS, automotive, government
20	Consultant	9	4 (4)	FS, government
21	Project/program manager	18	15 (15)	Telecommunication
22	Program manager	7	15 (10)	Telecommunication
	Ø	13.7	8.2 (5.9)	
Abbreviations: FS – Financial services; S&T – Sales and trading				

The interviews lasted between 90 and 120 minutes. All participants perceived RepGrid to be a useful, helpful, and pleasant interview technique. Many of the experts were amazed by the factors of which they became aware during the interviews. Accordingly, we are confident to have explored both explicit and tacit knowledge of our respondents.

Table 2. RepGrid Design Alternatives and our Choices

Design decision	Design alternatives	Our design choice
Conceptual direction	Research perspective: qualitative / quantitative	Qualitative
	RepGrid nature: idiographic / nomothetic	Idiographic
Logging data	Notes / recording & transcription	Recording & transcription
Identification of elements	Choosing the type of elements	EAI projects
	Supplying / eliciting elements	Eliciting elements
	Elements elicitation: roles / pool / discussion	Pool
	Number of elements	4
	Opposing elements demanded: yes / no	Yes
Identification of constructs	Supplying / eliciting constructs	Eliciting constructs
	Constructs elicitation: minimum context form (triads, dyads) / full context form / group elicitation	Dyads
	Elements comparison method: randomly / systematically	Systematically (all possible comparisons)
Laddering	Yes / no	Yes
	Downwards / sideways / upwards	Downwards
Linking elements to constructs	Yes / no	No
	Ranking / rating / dichotomizing	-

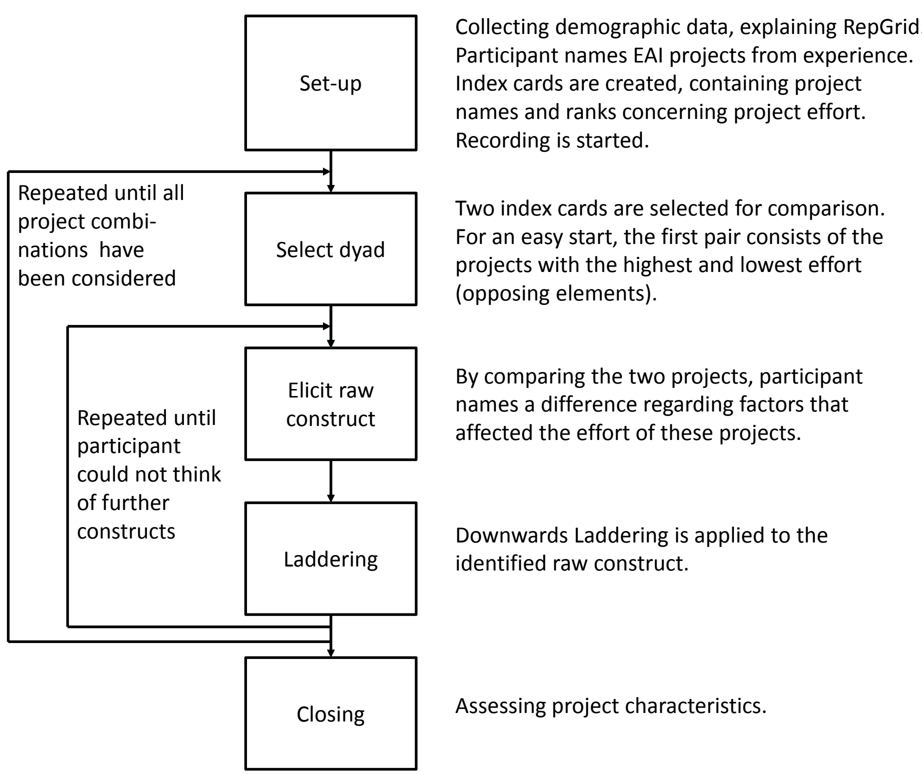

Figure 1. Interview Methodology

Data Analysis

We applied content analysis to the total of 422 raw constructs elicited in 22 interviews (minimum constructs per interview: 11, maximum: 31, median: 17, mean: 19). The objective of the analysis was to structure the elicited constructs and to identify commonalities and emphases across the respondents. This objective was achieved in three steps: (1) consolidating constructs that were mentioned multiple times, that is, duplicates, (2) grouping related constructs to construct sets, and (3) assigning the identified construct sets to categories. Figure 2 illustrates these steps and their outcomes.

(1) All 422 raw constructs were analyzed to develop an overall set of unique, that is, distinct constructs. This was achieved by consolidating duplicates. Only when two or more constructs meant the same according to construct card information and transcripts, we merged those constructs. In cases where such duplicates had (slightly) different names, one of them or a new name was chosen for the unique construct. We strived for a consistent terminology as well as at retaining the original wording in the total set of the emerging unique constructs. Before renaming a construct, we examined all construct-related information (in transcripts and on construct cards) to prevent change of meaning. For example, referring to the same concepts, some experts used the term integration tools while others preferred integration products (cf. *correspondence* as one of semantic constellations described in *Data Collection*). In such cases, we consistently used the same terms (here: integration products). This step yielded 310 unique constructs.

(2) We analyzed the 310 unique constructs and identified contentual emphases by grouping them to construct sets (of varying size) if there was a strong contentual relation. To these emerging construct sets, we assigned new denotations that captured the meaning of the included unique constructs. This first construct aggregation was also based on all available construct-related information to ensure accurate interpretation of construct meaning. Most emerged construct sets comprise two unique constructs (average 3.4; maximum 11). 13 unique constructs remained stand-alone, that is, they represent 13 construct sets. In the remainder of this paper, we refer to construct sets as *factors* (that influence the effort of EAI projects). Overall, 91 factors were identified in this step.

(3) We further aggregated by assigning the identified factors to categories. We did not predefine the categories; rather, they emerged in the process of reviewing the factors and arranging them according to content (Jankowicz 2004; Stewart et al. 1981). Since this is a subjective process, it is noteworthy that the resulting categories of a content analysis are not universally valid; categorizations based on the same data performed by other analysts could contain deviations. In total, we assigned the factors to 9 categories.

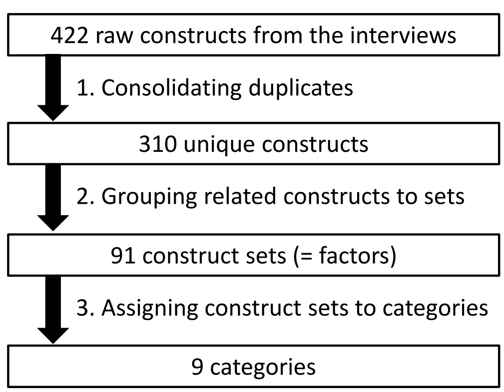

Figure 2. Content Analysis

As suggested by Stewart et al. (1981), two further researchers were involved in the aggregation process (steps 2 and 3) to increase reliability. These researchers were authors' colleagues working as graduates in the IS discipline. They reviewed both denotations (of factors and categories) and assignments (of unique constructs to factors and factors to categories). We modified few names and assignments considering their suggestions. Additionally, we sent the overall results to our respondents to ensure validity (Stewart et al. 1981; Whyte and Bytheway 1996). We marked constructs of the respective respondents in the overall scheme and asked them for review. All respondents approved our content analysis results.

Mapping of Factors to Previous Research

Contrasting our study to extant research – which predominantly addressed formal models for effort estimation (cf. *Effort Estimation of EAI Projects*) – we compare our factors to those included in formal models. Whereas a variety of models have been proposed since the 1960s and 1970s (Jørgensen et al. 2009), we focus on the three formal models that we identified to be designed for the estimation of application integration effort. We thus serve the purpose of covering a representative set of models for the context of our study. The identified models account for the peculiarities of integrating commercial off-the-shelf systems (Abts and Boehm 1997), system-of-system applications (Lane 2004), and systems engineering (Valerdi 2005). All three models are based on the Constructive Cost Model (CoCoMo) by Boehm et al. (2000), which is considered to be the most widely used estimation model for new software development (e.g., Valerdi and Madachy 2007). CoCoMo and CoCoMo-based models are used to estimate effort based on the size of software systems and multiple parameters (so-called cost drivers). While the former covers aspects like lines of code or number of use cases, the latter address characteristics like the project (e.g., use of software tools), personnel (e.g., team member experience), hardware (e.g., performance constraints), and product (e.g., product complexity). The parameters are assessed on a scale (e.g., from *very low* to *extra high*), thus affecting the estimate by multiplying the size of a software system by a value lower or higher than 1. For instance, software engineers with very low/extra high capabilities lead to an increase/decrease of the estimates, while a very low/extra high complexity of the systems leads to a decrease/increase of the estimate. Consequently, the parameters can be considered as effort-affecting factors. In our analysis, we accordingly link a formal model to a factor identified in our study if a parameter of that formal model matches that factor. To reduce subjectivity, two authors conducted the comparisons and discussed diverging assessments until agreement was reached. We provide the links of our factors to the three formal models in Table 4 in the *Results* section.

Quality Criteria

Following Glaser and Strauss (1967), a researcher "trying to discover theory cannot state at the outset of his research how many [participants] he will sample during the entire study; he can only count up the [participants] at the end" (p. 61). Accordingly, the numbers of RepGrid interviews considered appropriate in literature differ (Dillon and McKnight 1990; Dunn and Ginsberg 1986; Hassenzahl 2002). However, there is agreement among researchers that the desired objective is to reach theoretical saturation (Glaser and Strauss 1967; Strauss and Corbin 1990) regarding the elicited constructs. Theoretical saturation refers to the point at which no new findings emerge in subsequent interviews. In this context, Siau et al. (2010) speak of the 'point of redundancy', a term introduced by Lincoln and Guba (1985, p. 235). Striving for reaching this point, we analyzed the emerging constructs after sets of 2-3 interviews. As no new factors arose in the last three interviews (cf. Figure 3), we are confident to have reached theoretical saturation.

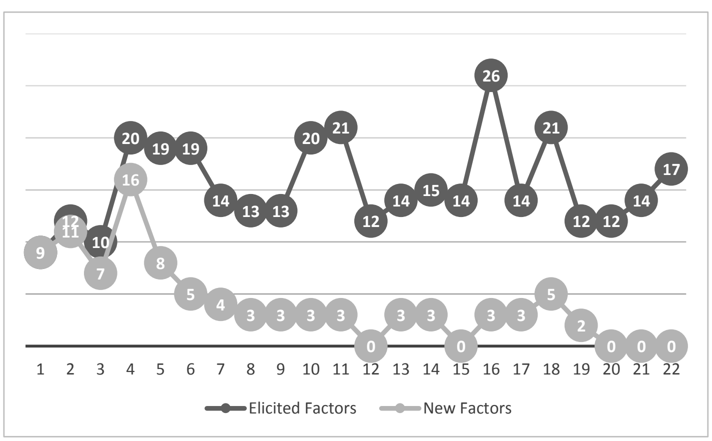

Figure 3. Theoretical Saturation (Factors per Interview)

To ensure communicative validity (Flick 2009), we sent the interview transcripts back to the respondents for verification. We ensured plausibility (Guba 1978; Patton 2002) of the categorization as follows. Two authors verified the conceptual sense in the categories (i.e., homogeneity within categories and heterogeneity among categories) and suggested minor changes. Having taken these steps makes us confident that we have provided a meaningful categorization in the context of factors affecting effort of EAI projects at this exploratory stage.

Results

As described above, we elicited 422 raw constructs in the 22 conducted interviews and consolidated them into 310 unique constructs. These unique constructs were grouped into 91 factors (=construct sets) that influence effort of EAI projects, which in turn were assigned to 9 categories. Table 3 provides an overview of the categories including their definitions, numbers of respondents who mentioned factors in the respective categories, as well as the numbers of factors and unique constructs contained in the categories.

Table 3. Identified Categories

No.	Category	Definition	# Respondents	# Factors / unique constructs
1.	Requirements	Requirements of the system that is developed to become the intended integration solution	20	15 / 55
2.	Systems' characteristics	Characteristics of the different application systems that are being integrated	20	12 / 35
3.	Design	All aspects concerning the design of the integration solution to be established	20	19 / 49
4.	Technology	All aspects of integration products and tools (availability, suitability, etc.) applied for the conversion of the different application systems into the integration solution	18	10 / 29
5.	Testing and rollout	Testing of the integration solution and its transition into usage	10	5 / 17
6.	Project team	All aspects regarding EAI project team members (i.e., the core team in charge of implementing the integration solution)	22	9 / 32
7.	External stakeholders	Further people involved in or affected by the integration project (i.e., all stakeholders except for the project team)	19	7 / 34
8.	Project management	All factors related to the management of the EAI project	12	4 / 22
9.	Conditions	Conditions resulting from project's environment	20	10 / 37

Table 4 lists all factors in their respective categories. In brackets, the two opposing values are given (representing the construct poles), whereby the first value is associated with lower and the second with higher effort. The column #UC provides the number of unique constructs grouped into the corresponding factors. Additionally, the three right columns show links of the factors to formal models (cf. *Research Design*). An 'x' means that the given factor is covered in the corresponding model. The models are referenced by numbers as follows: 1 – (Abts and Boehm 1997); 2 – (Lane 2004); 3 – (Valerdi, 2005). For each category, the amount of factors covered by at least one model is provided in the lowest row.

In the following, we first describe selected categories and included factors. We elaborate on the categories *1. Requirements*, *3. Design*, and *6. Project team* since the first two contain the most factors/unique constructs (category 1: 15/55 and category 3: 19/49) and category 6 is the only one that contains factors mentioned by all 22 respondents. In describing these categories, we focus on factors that comprised the highest number of unique constructs. Subsequently, we address the coverage of identified factors by formal models. Implications of these findings are discussed in the following section.

Table 4. Identified Factors Influencing Effort of EAI Projects					
No.	Factors	# UC	Models		
			1	2	3
1. Requirements					
1.1	General scope of the integration project (narrow vs. broad)	2			
1.2	Number, type, and complexity of affected business processes (low vs. high)	3			
1.3	Number and complexity of affected use cases (low vs. high)	2		x	x
1.4	Focus of the integration project (narrow vs. broad)	6			
1.5	Novelty of the integration solution (low vs. high)	5		x	
1.6	Novelty of affected business processes (low vs. high)	2			
1.7	Clarity of requirements definition (high vs. low)	9			x
1.8	Change dynamics of requirements (low vs. high)	5		x	x
1.9	Existence of an established change request system (yes vs. no)	1			
1.10	Quality-related requirements of the integration solution (low vs. high)	10	x		x
1.11	Requirements of monitoring the interaction between application systems (low vs. high)	2			
1.12	Number of users and user roles (low vs. high)	2			
1.13	Use distribution of the integration solution (central vs. distributed usage)	1			
1.14	Necessity of comparison of master data of affected application systems (no vs. yes)	3			
1.15	Necessity of a migration (no vs. yes)	2			
Amount of factors covered by at least one formal model: 5/15 = 33.3%					
2. Systems' characteristics					
2.1	Number and complexity of affected application systems (low vs. high)	2		x	x
2.2	Type of affected application systems (compatible vs. incompatible)	4			x
2.3	Availability of interfaces of affected application systems (yes vs. no)	3			
2.4	Suitability of interfaces for the integration solution (high vs. low)	2		x	
2.5	Quality of interfaces (high vs. low)	2			
2.6	Interfaces based on standards (yes vs. no)	1			
2.7	Change dynamics of interfaces (low vs. high)	1		x	
2.8	Necessity of adaption of affected application systems (no vs. yes)	1	x	x	
2.9	Heterogeneity of affected application systems (low vs. high)	5		x	x
2.10	System environment of affected application systems (consistent vs. inconsistent)	5			x
2.11	Quality of documentation and contact persons (high vs. low)	6			x
2.12	Operation of affected application systems (internal vs. external)	3			
Amount of factors covered by at least one formal model: 8/12 = 66.7%					
3. Design					
3.1	Investment in design and architecture of the integration solution (high vs. low)	3	x	x	
3.2	Complexity of integration architecture (low vs. high)	3	x		x
3.3	Change dynamics of integration architecture (low vs. high)	2		x	
3.4	Complexity of communication relations between affected application systems (low vs. high)	4			
3.5	Number and complexity of interfaces (low vs. high)	3	x	x	x
3.6	Complexity of adapters (low vs. high)	1			
3.7	Number of communication formats of affected application systems (low vs. high)	4			
3.8	Number and complexity of transformations (low vs. high)	2			

3.9	Complexity of data structures of the integration solution (low vs. high)	7			
3.10	An existing central (canonical) data model for transformations can be used (yes vs. no)	3			
3.11	Network boundaries to overcome with the integration solution (few vs. many)	2			
3.12	Design of communication between affected application systems (synchronous vs. asynchronous)	2			
3.13	Coupling between components of the integration solution (loose vs. fixed and vice versa)*	3		x	
3.14	Realization of interaction controls between affected application systems (no vs. yes)	2			
3.15	Number and complexity of modules of the integration solution (low vs. high)	2			
3.16	Number and complexity of functions to be implemented (low vs. high)	2		x	
3.17	Disposition of user authentication (no vs. yes)	1			
3.18	Considering business processes' scope during integration solution design (yes vs. no)	2			
3.19	Considering business processes' concurrency during integration solution design (yes vs. no)	1			
*: As indicated by the unique constructs grouped in this factor, both values can be associated with low and high effort, that is, loose coupling can lead to lower effort compared to fixed coupling and vice versa.					
Amount of factors covered by at least one formal model: 6/19 = 31.6%					
4. Technology					
4.1	Availability of integration products (yes vs. no)	4			
4.2	Challenge of understanding and using applied integration products (low vs. high)	2	x		x
4.3	Quality of applied integration products (high vs. low)	4	x		
4.4	Change dynamics of applied integration products (low vs. high)	2			
4.5	Suitability of applied integration products for the integration solution (high vs. low)	5			
4.6	Dependency on manufacturers of applied integration products (no vs. yes)	2			
4.7	Support of manufacturers of applied integration products (yes vs. no)	1	x		
4.8	Number and complexity of applied technologies (low vs. high)	5			x
4.9	Employment of development tools (yes vs. no)	3	x	x	x
4.10	Compliance with standards of affected application systems (yes vs. no)	1			
Amount of factors covered by at least one formal model: 5/10 = 50%					
5. Testing and rollout					
5.1	Extent of testing (low vs. high)	3			
5.2	Rigor of testing (high vs. low)	4			
5.3	Prevailing testing conditions (simple vs. complicated)	6			
5.4	Distribution of testing responsibilities (clear vs. unclear)	2			
5.5	System rollout approach (simple vs. complex)	2			
Amount of factors covered by at least one formal model: 0/5 = 0%					
6. Project team					
6.1	Number of team members (low vs. high)	1			
6.2	Productivity of team members (high vs. low)	1			x
6.3	Homogeneity of team members (high vs. low)	3			x
6.4	Experience and knowledge of team members (high vs. low)	11	x	x	x
6.5	Quality of cooperation between team members (high vs. low)	3		x	x
6.6	Familiarity of project team (well-established vs. newly composed)	3	x	x	x
6.7	Team members' motivation related to the integration project (high vs. low)	4			
6.8	Team collaboration with other organizations during development (no vs. yes and vice versa)*	4			

6.9	Quality of communication between team members (high vs. low)	2		x	
*: Both values can be associated with low and high effort, that is, in-house development might result in lower effort compared to cooperation with other organizations and vice versa.					
Amount of factors covered by at least one formal model: 6/9 = 66.7%					
7. External stakeholders					
7.1	Delineation of the relationship with customer (clear vs. unclear)	5			x
7.2	Stakeholder homogeneity within the integration project (high vs low)	4		x	x
7.3	Stakeholders' attitude towards the integration project (positive vs. negative)	4			
7.4	Communication with customer (direct, formal vs. indirect, informal)	4			
7.5	Extent of support provided by customer (high vs. low)	5			
7.6	Extent of customer experience and knowledge (high vs. low)	6			
7.7	Responsibilities on the part of customer (clear vs. unclear)	6			
Amount of factors covered by at least one formal model: 2/7 = 28.6%					
8. Project management					
8.1	Quality of management of the integration project (high vs. low)	9		x	x
8.2	Extent of proactive approach (high vs. low)	2			
8.3	Design of the integration project organization (clear vs. unclear)	7			x
8.4	Applied development approach (sequential vs. evolutionary and vice versa)*	4		x	
*: Both values can be associated with low and high effort, that is, a sequential development approach might result in lower effort compared to an evolutionary approach and vice versa.					
Amount of factors covered by at least one formal model: 3/4 = 75%					
9. Conditions					
9.1	Duration of the integration project (low vs. high)	2			
9.2	Novelty of the integration project (low vs. high)	1			
9.3	Clarity of objectives of the integration project (high vs. low)	5			
9.4	Relevance of the integration project for customer (low vs. high)	3		x	
9.5	Extent of internationality of the integration project (low vs. high)	4			x
9.6	Spatial distribution of the project stakeholders (local vs. distributed)	2			x
9.7	Availability of suitable infrastructure for conducting the integration project (yes vs. no)	4		x	
9.8	Dependency of the integration project on other projects (low vs. high)	5			
9.9	Guidelines for development (vague vs. strict)*	8			
9.10	Available budget/timeframe (small/tight vs. large/broad)	3		x	
*: Both values can be associated with low and high effort, that is, vague guidelines for development might result in lower effort compared to strict demands and vice versa.					
Amount of factors covered by at least one formal model: 5/10 = 50%					
Total amount of factors covered by at least one formal model: 40/91 = 44%					
	Total	310	11	24	25

Within the category *1. Requirements*, several emphases can be identified. The most (10) unique constructs were grouped into the factor *1.10 Quality-related requirements of the integration solution*. Our respondents described that higher quality requirements consistently lead to higher effort. In particular, requirements concerning safety, availability, and performance of the integration solution were mentioned in this regard. Moreover, 9 unique constructs were grouped into *1.7 Clarity of requirements definition*. In the view of our respondents, project effort was higher if requirements were not clearly defined, that is, the integration problem and its intended solution were insufficiently analyzed and/or documented. Similarly, effort increased if projects showed higher *change dynamics of requirements (*1.8; 5 unique constructs*)* compared to stable scope. The *focus of the integration project* (1.4; 6 unique constructs) comprises various effort-influencing aspects with regard to project direction. For example, the effort was lower if the integration endeavor was focused on technical aspects compared to projects with technical *and* business emphases. Similarly, effort was lower if the integration was information-oriented rather than process-oriented, or if business processes were not to be fully automated. *1.5 Novelty of the integration solution* (5 unique constructs) implies that project effort increases with higher degree of system's novelty. For example, our respondents stated that it is less complex to adapt/extend an existing system compared to developing a new integration solution from scratch.

In category *3. Design, 3.9 Complexity of data structures of the integration solution* comprises the most (7) unique constructs. Effort increases with higher complexity of data structure, ranging from the complexity of the underlying data model, over database tables, to single data elements. Another emphasis lies on the communication between the application systems that are being integrated. 4 unique constructs were merged into the factors *3.4 Complexity of communication relations between affected application systems* and *3.7 Number of communication formats of affected application systems*, respectively. While the former refers to structural communication aspects like one-to-one vs. many-to-many or unidirectional vs. bidirectional communication relations between systems, the latter considers the number of data records and message formats. For all described factors, higher number and complexity result in increased effort.

Finally, all 22 respondents stated at least one factor in category *6. Project team*. The factor with by far the most unique constructs (11) in this category is *6.4 Experience and knowledge of team members*. Both general and various subject-specific experiences of team members were emphasized. Available experienced team members expectably led to reduced project effort. Furthermore, the *team members' motivation related to the integration project* (6.7; 4 unique constructs) was seen as an important influence on project effort, in particular their willingness to face challenges and novel characteristics of application integration. For instance, our respondents explained that effort decreases if team members do not insist on working in familiar ways and with specific well-known technologies only.

In four cases (factors 3.13, 6.8, 8.4, and 9.9), both factor values were said to possibly lead to lower or higher effort. In other words, the poles of the unique constructs grouped in the respective factors were opposed. For instance, considering *8.4 Applied development approach*, one respondent stated a sequential development to lead to lower effort compared to an evolutionary one, while another respondent recalled the opposite experience. This is not surprising since both approaches have their advantages and disadvantages, influencing effort accordingly.

The highest amount of factors matched by at least one formal model is found in the category *8. Project management* (75%). The categories *2. Systems' characteristics* and *6. Project team* follow with 66.7%, respectively. 50% are covered in *4. Technology* and *9. Conditions*. Close together lie the categories *1. Requirements* (33.3%), *3. Design* (31.6%), and *7. External stakeholders* (28.6%). Finally, none of the factors in *5. Testing and rollout* is found in the formal models. Moreover, it is noticeable that the majority of factors are matched by the two more recent models. While models 2 (Lane 2004) and 3 (Valerdi, 2005) cover 24 and 25 factors, respectively, and 38 factors in total, model 1 (Abts and Boehm 1997) includes 11 factors, of which only two (4.3 and 4.7) are not covered by models 2 or 3. We discuss our findings below.

Discussion

Our study's major outcome are nine categories containing 91 factors that influence the effort of EAI projects. This overview helps to better understand EAI projects and provides a useful guideline to assess the variety of effort-influencing factors that should be considered in effort estimation of such projects. Hence, the list of factors can be used to provide more accurate and reliable effort estimates. Below, we discuss implications of our study, describe its limitations, and provide guidelines for future research.

Implications

Following research on effort estimation in general (Jørgensen et al. 2009), a major differentiation concerns the choice between formal models and expert judgments. The variety and number of unique constructs (310) and factors (91) within the nine categories show that formal models for accurate effort estimation of EAI projects would need to cover a multitude of effort drivers, making the suitability of such models in general rather questionable. Our results thus contribute to the research stream promoting the role of expert judgments for effort estimation. Practitioners can use our overview as a checklist for expert judgments by reviewing it and deciding which factors are relevant in given projects. Experts can thus ensure to include all relevant aspects in their estimates, including more and less intuitive factors.

Nevertheless, researchers can apply our overview to improve existing models. The mapping of our factors to parameters of models for effort estimation (cf. Table 4) reveals several interesting insights. Existing models include factors of at least five (model 1) and at most eight (models 2 and 3) categories in our study, spanning a wide range of project themes in general. However, the included parameters do not sufficiently cover the variety of factors revealed in this exploratory research. Overall, only 40 of the 91 identified factors (44%) have a counterpart in at least one estimation model, which corroborates Rosa et al.'s (2013) assessment that existing formal estimation models do not suit the requirements of effort estimation of EAI projects. The remaining 51 identified factors that are not covered in the existing models reveal multiple factors that are rather intuitive, for example *1.2 Number, type, and complexity of affected business processes*, *2.2 Type of affected application systems*, and *9.9 Guidelines for development*. Nevertheless, such factors need to be included in effort estimation, and the fact that they are neglected in existing models is alarming. Enriching those models with newly identified factors, however, involves the challenge of distinguishing between the critical and less important factors. Including all aspects seems unfeasible since assessing all these parameters would present a major challenge for project team members – as integration projects are typically highly complex, it is unlikely that all required information is available. In the following, we thus discuss the emerged categories and identify emphases regarding the factors that are not yet covered by estimation models.

In category *1. Requirements*, the factor *1.12 Number of users and user roles* is in our view of particular importance. Change management literature highlights addressing stakeholder resistance in an effective way as critical for project success (Hirschheim and Newman 1988; Keen 1981). Since this endeavor becomes more complicated with a growing number of stakeholders, numerous user roles are likely to considerably increase the required change management effort. Similarly, the *number, type, and complexity of affected business processes (1.2)* might have extensive impact on change management as changes of business processes result in new and unfamiliar tasks, causing resistance by users. The importance of these factors is further intensified by the finding that the interoperability of affected systems impacts integration effort most substantially at the business level (Mouzakitis et al. 2009).

Since category *2. Systems' characteristics* shows a high coverage of factors by formal models (66.7%), we believe that it is in general well represented in those models. However, additional attention should be paid to the applied interfaces, which are subject of three (2.3, 2.5, and 2.6) out of four not covered factors. Effort might increase substantially if, for instance, such interfaces are not available (2.3) or prove qualitatively inferior (2.5) for the intended integration solution. Such difficulties might be amplified by existing strict guidelines which interfaces to use (cf. Discussion of factor 9.9 below).

In *3. Design,* only six out of 19 factors are found in estimation models (31.6%). This is surprising since several remaining factors seem intuitive (e.g., *3.4 Complexity of communication relations between affected application systems*, *3.7 Number of communication formats of affected application systems*, and *3.9 Complexity of data structures of the integration solution*; all three factors are described in the *Results* section). In our view, one emphasis in this category relates to the communication between affected application systems, which is addressed by several factors (3.4, 3.7, and 3.12). However, we believe that all remaining factors should receive attention in effort estimation considering that this category contains the most factors, of which only 31.6% are found in formal models.

In *4. Technology,* four (4.1, 4.4, 4.5, and 4.6) out of the five factors not considered in the three formal models are concerned with integration products. These are standard components of the integration solution, which support the interaction of the individual systems and can usually be acquired externally. Along the lines of interfaces in category 2, effort is likely to increase if, for instance, such products are not available (4.1) or prove unsuitable (4.5) for the intended integration solution. We thus stress the importance of considering integration products for effort estimation, especially in case of strict guidelines which products to use (cf. Discussion of factor 9.9 below).

None of the factors in category *5. Testing and rollout* is accounted for in existing models. This is surprising since testing and rollout are essential phases in system development projects (Reel 1999; Sommerville 2011). Additionally, quality assurance endeavors are known to be regularly underestimated and to result in a multiple of the planned effort (Stelzer et al. 1997). We thus assess the entire category as critical and advise to account for included factors in effort estimation.

In *6. Project team,* we highlight *6.7 Team members' motivation related to the integration project* as it is likely to influence other factors. While *6.4 Experience and knowledge of team members* is obviously important (11 unique constructs and included in all three models), team members might not exploit their full potential if being unmotivated in the first place; productivity (6.2) and quality of cooperation/ communication (6.5/6.9) are likely to decline in this case as well. Our findings are thus complementary and in agreement with extant research reporting that motivation has a substantial impact on productivity (Beecham et al. 2008; Boehm 1981), yet continues to receive too little attention in practice (Beecham et al. 2008; Procaccino et al. 2005). Moreover, factor *6.8 Team collaboration with other organizations during development* should not be neglected. While most included unique constructs suggest that necessity of such collaboration increases effort, our experts also mentioned the possibility of reduced effort due to shared responsibility. Accordingly, impact of collaboration needs to be examined in concrete situations.

In *7. External stakeholders,* the factor *7.3 Stakeholders' attitude towards the integration project* needs special attention. Along the lines of team members' motivation as discussed in category 6, the attitude of other stakeholders is likely to influence further factors within category 7, for instance, the extent of support provided by the customer for the project (7.5). However, since this is the second-least covered category (28.6%) after *5. Testing and rollout,* we believe the entire category should be carefully considered when estimating EAI project effort. Thereby, the customer carries particular weight as an external stakeholder, which is evidenced by five factors explicitly concerned with this stakeholder group.

In *8. Project management,* the only factor not covered by an existing model is *8.2 Extent of proactive approach*, which refers to a positive effect of anticipatory thinking. Overall, this category is in our view the least critical one as it contains the fewest factors, 75% of which are included in existing models.

In *9. Conditions, 9.9 Guidelines for development* comprises the most unique constructs and is not matched by a formal model. Both values of this factor can be associated with lower/higher effort: While some experts reported that specific customer demands regarding the development (e.g., which interfaces or integration products to use) led to higher effort, others described the opposite effect. A relation to other identified factors (cf. Discussion of categories 2 and 4) further emphasizes the importance of this factor. Another factor that we highlight in this category is *9.8 Dependency of the integration project on other projects*. In an independent EAI project, effort estimation is already a complex undertaking. However, if coordination with other projects (which produce necessary inputs) is needed, the required effort becomes considerably more difficult to predict since the uncertainty of further projects is added to the equation.

Limitations and Future Research

As with any study, there are some limitations that need to be acknowledged. While identifying emphases among factors that are not considered in formal models, our results do not provide the relative importance of all factors affecting the effort of EAI projects. Although it can be assumed that the number of unique constructs indicates the importance of factors, future research should aim to systematically identify the factors with the greatest impact on effort of EAI projects. Several interviewees suggested that factors relevant for the development of new applications are differently important for the effort of EAI projects. In assessing the relative importance of factors in more detail, future research might also analyze whether the factors' importance is contingent on characteristics of EAI projects. Due to the qualitative nature of our study, we are not able to provide this kind of analysis. Moreover, assuming factors to be hubs in a cause-and-effect network, specific paths between factors and resulting effort might exist in EAI projects. This raises the question of interdependencies between the identified factors. Unfortunately, our findings do not reveal such interdependencies. While the focus of our study was the identification of factors, we suggest future research to address factor chains. Uncovering these chains, researchers should be able to identify factors that are pivotal to the chains and, accordingly, more important to the effort of EAI projects than others. Moreover, the elicited factors can be seen as hypotheses stating that the respective factors influence the effort of EAI projects and should be tested in future research. Yet, our results are a crucial (first) step for a comprehensive understanding of factors influencing the effort of EAI projects. Further enrichment of knowledge about the factors will contribute to a better understanding in this domain of project management. Following this purpose might also reveal insights that have not been gained within this study. Nevertheless, RepGrid has been shown to be a suitable means to identify the factors since several respondents claimed that unconscious knowledge has been elicited, confirming the RepGrid's ability to transform tacit knowledge into explicit knowledge. Finally, as regards the comparison of our results with previous works, we acknowledge that partially abstract descriptions of related objects of investigation, definition of terms, and assumptions impeded the mapping of model parameters to the factors in our study. To address this issue, two authors scrutinized the models and discussed diverging assessments to ensure a suitable comparison. We thus believe the mapping of factors to model parameters to be adequate to provide a general assessment of the suitability of effort estimations models for EAI projects.

Conclusion

Our study provides empirical insights into factors that impact the effort of EAI projects and thus need to be considered in effort estimation of such projects. Despite aforementioned limitations, this paper contributes to a deeper understanding of effort of EAI projects. The comparison of our results with previous research, more concretely, existing formal models for effort estimation, reveals a multitude of factors that have not been considered yet. Neglecting those factors when estimating the effort of EAI projects is a likely explanation for the high time and budget overruns that are typical in such projects. Using our overview as a checklist for effort estimation based on expert judgments, practitioners can make sure to include all relevant aspects in their estimates. Furthermore, researchers can use our overview to enhance existing models by enriching them with factors identified in this study. With the exploration of the factors and their classification, we lay the groundwork for future studies and advance existing insights into the effort of EAI projects.

Appendix A

Table 5. Company Characteristics

Industry	Software development (20%)	IT service provider (80%)	
Type	Independent company (70%)	Subsidiary (30%)	
Customers	Internal (20%)	External (80%)	
	Min	Mean	Max
Annual revenue (MM EUR)	6	15,760	96,500
# Employees	40	55,584	319,000

Appendix B

Table 6. Project Characteristics

Duration in month	Up to 12 (62%)	13-24 (21%)	25-36 (11%)	More than 36 (6%)	
Scope of integration	Intra-divisional (12%)	Inter-divisional (36%)	Intra-organizational (38%)	Inter-organizational (14%)	
Internationality	National (70%)	International (30%)			
Type of integrated systems (multiple select)	ERP system (49%)	Legacy system (60%)	Other standard system (54%)	Other individual system (79%)	Internet-based application (48%)
Novelty of integration solution	New development (67%)	Adaption (23%)	Maintenance (10%)		
Respondent's responsibility (multiple select)	Project manager (56%)	Architect (41%)	Developer (26%)	Planning / estimating (4%)	Other (11%)

References

Abts, C. M., and Boehm, B. W. 1997. *COTS Software Integration Cost Modeling Study. USC Technical Report*, Los Angeles: University of Southern California, Center for Software Engineering.

Basten, D., and Sunyaev, A. 2014. "A Systematic Mapping of Factors Affecting Accuracy of Software Development Effort Estimation," *Communications of the Association for Information Systems* (34:Article 4), pp. 51–86.

Beecham, S., Baddoo, N., Hall, T., Robinson, H., and Sharp, H. 2008. "Motivation in Software Engineering: A Systematic Literature Review," *Information and Software Technology* (50:9-10), pp. 860–878.

Boehm, B. W. 1981. *Software Engineering Economics*, Englewood Cliffs: Prentice-Hall.

Boehm, B. W., Abts, C., Brown , A. Winsor, Chulani, S., Clark, B. K., Horowitz, E., Madachy, R., Reifer, D. J., and Steece, B. 2000. *Software Cost Estimation with CoCoMo II*, Upper Saddle River: Prentice Hall.

Boehm, B. W., and Fairley, R. E. 2000. "Software Estimation Perspectives," *IEEE Software* (17:6), pp. 22–26.

Boehm, B. W., and Valerdi, R. 2008. "Achievements and Challenges in Cocomo-Based Software Resource Estimation," *IEEE Software* (25:5), pp. 74-83.

Connolly, T., and Dean, D. 1997. "Decomposed versus Holistic Estimates of Effort Required for Software Writing Tasks," *Management Science* (43:7), pp. 1029–1045.

Crosman, P., Bruno-Britz, M., and O'Donnell, A. 2007. "2008 IT Budgets Up More Than 10% for Financial Services Firms," Wall Street & Technology.

Daneva, M. 2008. "Complementing Approaches in ERP Effort Estimation Practice: An Industrial Study," in *Proceedings of the 4th international workshop on Predictor models in software engineering*, Leipzig, Germany: ACM, pp. 87–92.

Daneva, M. 2010. "Balancing Uncertainty of Context in ERP Project Estimation: An Approach and a Case Study," *Journal of Software Maintenance and Evolution: Research and Practice* (22:5), pp. 329–357.

Dejaeger, K., Verbeke, W., Martens, D., and Baesens, B. 2012. "Data Mining Techniques for Software Effort Estimation: A Comparative Study," *IEEE Transactions on Software Engineering* (38:2), pp. 375–397.

Dillon, A., and McKnight, C. 1990. "Towards a Classification of Text Types: A Repertory Grid Approach," *International Journal of Man-Machine Studies* (33:6), pp. 623–636.

Dunn, W. N., and Ginsberg, A. 1986. "A Sociocognitive Network Approach to Organizational Analysis," *Human Relations* (39:11), pp. 955–975.

Flick, U. 2009. *An Introduction to Qualitative Research*, Los Angeles: Sage.

Francalanci, C. 2001. "Predicting the Implementation Effort of ERP Projects: Empirical Evidence on SAP/R3," *Journal of Information Technology* (16:1), pp. 33–48.

Fransella, F., Bell, R., and Bannister, D. 2004. *A Manual for Repertory Grid Technique*, Chichester, West Sussex, England: John Wiley & Sons.

Ganly, D. 2009. "Estimating the Costs of an ERP/Business Application Initiative," Gartner RAS Core Research Note G00167162.

Gartner Inc. 2006. "Gartner Says Worldwide Application Integration and Middleware Market Increased 7 Percent in 2005," Gartner Inc. (ed.), Stamford, CN.

Gartner Inc. 2008. "Gartner Says Worldwide Application Infrastructure and Middleware Market Revenue Increased 13 Percent in 2007," I. Gartner (ed.), Stamford, CN.

Gericke, A., Klesse, M., Winter, R., and Wortmann, F. 2010. "Success Factors of Application Integration: An Exploratory Analysis," *Communications of the Association for Information Systems* (27:Article 37), pp. 677–694.

Glaser, B. G., and Strauss, A. L. 1967. *The Discovery of Grounded Theory. Strategies for Qualitative Research*, Chicago: Aldine Pub. Co.

Gray, A. R., MacDonell, S. G., and Shepperd, M. 1999. "Factors Systematically Associated with Errors in Subjective Estimates of Software Development Effort: The Stability of Expert Judgment," in *Proceedings of the Sixth International Software Metrics Symposium*, IEEE Computer Society (ed.), New York: IEEE Computer Society Press, pp. 216–227.

Grimstad, S., and Jørgensen, M. 2007. “Inconsistency of Expert Judgment-based Estimates of Software Development Effort,” *Journal of Systems and Software* (80:11), pp. 1770–1777.
Grimstad, S., Jørgensen, M., and Moløkken-Østvold, K. 2006. “Software Effort Estimation Terminology: The Tower of Babel,” *Information and Software Technology* (48:4), pp. 302–310.
Gruschke, T. M., and Jørgensen, M. 2008. “The Role of Outcome Feedback in Improving the Uncertainty Assessment of Software Development Effort Estimates,” *ACM Transactions on Software Engineering and Methodology* (17:4), pp. 20:1-20:35.
Guba, E. G. 1978. *Toward a Methodology of Naturalistic Inquiry in Educational Evaluation*, Los Angeles, USA: University of California.
Hasselbring, W. 2000. “Information System Integration,” *Communications of the ACM* (43:6), pp. 32-38.
Hassenzahl, M. 2002. “Character Grid: A Simple Repertory Grid Technique for Web Site Analysis and Evaluation,” in *Human Factors and Web Development*, J. Ratner (ed.), Mahwah: Lawrence Erlbaum Associates, Inc., pp. 183–206.
Heemstra, F. J., and Kusters, R. J. 1991. “Function Point Analysis: Evaluation of a Software Cost Estimation Model,” *European Journal of Information Systems* (1:4), pp. 229–237.
Hinkle, D. N. 1965. *The Change of Personal Constructs from the Viewpoint of a Theory of Construct Implications. PhD Dissertation*, Ohio: Ohio State University.
Hirschheim, R., and Newman, M. 1988. “Information Systems and User Resistance: Theory and Practice,” *The Computer Journal* (31:5), pp. 398–408.
Ho, L. T., and Lin, G. C. I. 2004. “Critical Success Factor Framework for the Implementation of Integrated-enterprise Systems in the Manufacturing Environment,” *International Journal of Production Research* (42:17), pp. 3731–3742.
Jankowicz, D. 2001. “Why Does Subjectivity Make Us Nervous? Making the Tacit Explicit,” *Journal of Intellectual Capital* (2:1), pp. 61–73.
Jankowicz, D. 2004. *The Easy Guide to Repertory Grids*, Chichester: Wiley.
Jørgensen, M. 2004a. “A Review of Studies on Expert Estimation of Software Development Effort,” *Journal of Systems and Software* (70:1-2), pp. 37–60.
Jørgensen, M. 2004b. “Top-down and Bottom-up Expert Estimation of Software Development Effort,” *Information and Software Technology* (46:1), pp. 3–16.
Jørgensen, M. 2014. “What We Do and Don't Know about Software Development Effort Estimation,” *IEEE Software* (31:2), pp. 37–40.
Jørgensen, M., and Boehm, B., and Rifkin, S. 2009. “Software Development Effort Estimation: Formal Models or Expert Judgment?” *IEEE Software* (26:2), pp. 14–19.
Keen, P. G. W. 1981. “Information Systems and Organizational Change,” *Communications of the ACM* (24:1), pp. 24-33.
Kelly, G. A. 1955. *The psychology of personal constructs*, New York: Norton.
Kemerer, C. F. 1987. “An Empirical Validation of Software Cost Estimation Models,” *Communications of the ACM* (30:5), pp. 416–429.
Khoumbati, K., and Themistocleous, M. 2006. “Integrating the IT Infrastructures in Healthcare Organisations: A Proposition of Influential Factors,” *Electronic Journal of e-Government* (4:1), pp. 27–36.
Kugeler, M., and Vieting, M. 2003. “Design of a Process-Oriented Organizational Structure,” in *Process Management: A Guide for the Design of Business Processes*, J. Becker, M. Kugeler, and M. Rosemann (eds.), Berlin: Springer, pp. 165–205.
Lam, W. 2005. “Investigating Success Factors in Enterprise Application Integration: A Case-driven Analysis,” *European Journal of Information Systems* (14:2), pp. 175–187.
Lane, J. A. 2004. “Constructive Cost Model for System-of-System Integration,” in *ACM-IEEE International Symposium on Empirical Software Engineering*. August.
Lincoln, Y. S., and Guba, E. G. 1985. *Naturalistic inquiry*, Newbury Park: Sage.
Linthicum, D. S. 2000. *Enterprise Application Integration*, Reading: Addison-Wesley.
Linthicum, D. S. 2004. *Next Generation Application Integration: From simple Information to Web Services*, Boston: Addison-Wesley.
Lublinsky, B., and Farrell, J. M. 2002. “Top 10 Reasons Why EAI Fails,” *EAI Journal* (12), pp. 41–42.
Markus, M. L. 2000. “Paradigm Shifts - E-Business and Business/Systems Integration,” *Communications of the Association for Information Systems* (4:Article 10), pp. 1–45.

Mendoza, L. E., Pérez, M., and Grimán, A. 2006. "Critical Success Factors for Managing Systems Integration," *Information Systems Management* (23:2), pp. 56–75.
Moløkken-Østvold, K., Jørgensen, M., Tanilkan, S. S., Gallis, H., Lien Anette C., and Hove Siw E 2004. "A survey on software estimation in the norwegian industry," in *Proceedings of the 10th International Symposium on Software Metrics*, IEEE Computer Society (ed.), Chicago. 11.-17. September 2004, New York: IEEE Computer Society Press, pp. 208–219.
Mouzakitis, S., Sourouni, A.-M., and Askounis, D. 2009. "Effects of Enterprise Interoperability on Integration Efforts in Supply Chains," *International Journal of Electronic Commerce* (14:2), pp. 127–155.
Napier, N. P., Keil, M., and Tan, F. B. 2009. "IT Project Managers' Construction of Successful Project Management Practice: A Repertory Grid Investigation," *Information Systems Journal* (19:3), pp. 255–282.
Nonaka, I. 1994. "A Dynamic Theory of Organizational Knowledge Creation," *Organization Science* (5:1), pp. 14–37.
Patton, M. Q. 2002. *Qualitative Research & Evaluation Methods*, Thousand Oaks: Sage.
Polanyi, M. 1966. *The Tacit Dimension*, Chicago, London: University of Chicago Press.
Procaccino, J. D., Verner, J. M., Shelfer, K. M., and Gefen, D. 2005. "What do Software Practitioners Really Think about Project Success: An Exploratory Study," *Journal of Systems and Software* (78:2), pp. 194–203.
Reel, J. S. 1999. "Critical Success Factors in Software Projects," IEEE Software (16:3), pp. 18–23.
Riempp, G., and Gieffers-Ankel, S. 2007. "Application Portfolio Management: A Decision-oriented View of Enterprise Architecture," *Information Systems and e-Business Management* (5:4), pp. 359–378.
Rosa, W., Packard, T., Krupanand, A., Bilbro, J. W., and Hodal, M. M. 2013. "COTS Integration and Estimation for ERP," *Journal of Systems and Software* (86:2), pp. 538–550.
Rugg, G., Eva, M., Mahmood, A., Rehman, N., Andrews, S., and Davies, S. 2002. "Eliciting information about organizational culture via laddering," *Information Systems Journal* (12:3), pp. 215–229.
Ruh, W. A., Maginnis, F. X., and Brown, W. J. 2000. *Enterprise Application Integration. A Wiley Tech Brief*, New York: John Wiley.
Schwinn, A., and Winter, R. 2007. "Success Factors and Performance Indicators for Enterprise Application Integration," in *Enterprise Architecture and Integration*, W. Lam, and V. Shankararaman (eds.), Hershey: IGI Global, pp. 23–39.
Seaver, D. 2006. "Estimating Enterprise Resource Planning," Soft Ware Engineering Institute SEPG.
Shaw, M. L., and Gaines, B. R. 1989. "Comparing conceptual structures: consensus, conflict, correspondence and contrast," *Knowledge Acquisition* (1:4), pp. 341–363.
Siau, K., Tan, X., and Sheng, H. 2010. "Important Characteristics of Software Development Team Members: An Empirical Investigation using Repertory Grid," *Information Systems Journal* (20:6), pp. 563–580.
Smith, M. 1986. "A Repertory Grid Analysis of Supervisory Jobs," *Applied Psychology* (35:4), pp. 501–511.
Sommerville, I. 2011. *Software engineering*, Boston: Pearson.
Sprott, D. 2000. "Enterprise Resource Planning: Componentizing the Enterprise Application Packages," *Communications of the ACM* (43:4), pp. 63–69.
Stelzer, D., Mellis, W., and Herzwurm, G. 1997. "A Critical Look at ISO 9000 for Software Quality Management," *Software Quality Journal* (6:2), pp. 65–79.
Stensrud, E. 2001. "Alternative Approaches to Effort Prediction of ERP Projects," *Information and Software Technology* (43:7), pp. 413–423.
Stensrud, E., and Myrtveit, I. 1998. "Human Performance Estimating with Analogy and Regression Models: An Empirical Validation," in *IEEE International Software Metrics Symposium*, IEEE Computer Society (ed.), Bethesda, M, Washington, DC, pp. 205–213.
Stewart, V., Stewart, A., and Fonda, N. 1981. *Business Applications of Repertory Grid*, London: McGraw-Hill Book Co.
Strauss, A. L., and Corbin, J. M. 1990. *Basics of Qualitative Research. Techniques and Procedures for Developing Grounded Theory*, Newbury Park: SAGE Publications.
Tan, F., and Gallupe, R. 2006. "Aligning business and information systems thinking: a cognitive approach," *IEEE Transactions on Engineering Management* (53:2), pp. 223–237.
Tan, F. B., and Hunter, M. G. 2002. "The Repertory Grid Technique: A Method for the Study of Cognition in Information Systems," *MIS Quarterly* (26:1), pp. 39–57.

Themistocleous, M. 2004. "Justifying the Decisions for EAI Implementations: A Validated Proposition of Influential Factors," *Journal of Enterprise Information Management* (17:2), pp. 85–104.
Themistocleous, M., and Irani, Z. 2002. "Novel Taxonomy for Application Integration," *Benchmarking: An International Journal* (9:2), pp. 154–165.
Thompson, J. 2007. "Integration Is Out of the Limelight But Growing in Importance," ID Number: G00145417, Gartner Inc., Stamford, CN.
Tuft, B. 2001. "The Changing Role of IT Strategy: Enterprise Architecture Strategies," *EAI Journal* (3:1).
Valerdi, R. 2005 "The Constructive Systems Engineering Cost Model (COSYSMO)", Dissertation, University of Southern California.
Valerdi, R., and Madachy, R. 2007. "Impact and Contributions of MBASE on Software Engineering Graduate Courses," *Journal of Systems and Software* (80:8), pp. 1185–1190.
Whyte, G., and Bytheway, A. 1996. "Factors Affecting Information Systems' Success," *International Journal of Service Industry Management* (7:1), pp. 74–93.
Xia, W., Ho, D., and Capretz, L. F. 2008. "A Neuro-fuzzy Model for Function Point Calibration," *WSEAS Transactions on Information Science and Applications* (5:1), pp. 22–30.
Yang, J., and Papazoglou, M. P. 2000. "Interoperation Support for Electronic Business," *Communications of the ACM* (43:6), pp. 39–47.
Zahavi, R. 2000. *Enterprise Application Integration with CORBA. Component and Web-based Solutions*, New York: Wiley.
Zaitun, A. B. 2001. "Success Factors to Systems Integration Implementation: More Technically Oriented than Human Related," *Malaysian Journal of Computer Science* (14:2), pp. 64–69.